CLAIMS

1. A method for multiplexing control signals for disk drives comprising:

developing parallel control signals;

developing serial control signals;

coupling at least one of the parallel control signals and the serial control signals to at least one of a parallel hard disk drive and a serial hard disk drive by a common control bus.

2. A method as recited in claim 1 wherein the first the parallel hard disk drive is an ATA type.

3. A method as recited in claim 1 wherein the serial hard disk drive is an SATA type.

4. A method as recited in claim 1 further comprising a second serial hard disk drive.

5. A method as recited in claim 4 wherein data is sent to the serial hard disk drive and the second serial hard disk drive at effectively double a base data rate.

6. A method as recited in claim 5 wherein the doubling the base data rate comprises:

developing a sampling data clock;

developing a first data stream at the base data rate;

developing a second data stream at the base data rate; and

multiplexing the first data stream to the common control bus on a rising edge of the base data clock and the second data stream to the common control bus on a falling edge of the base data clock, whereby the common control bus carries both the first data stream and the second data stream at effectively double the base data rate.

7. A method as recited in claim 1 for encoding additional commands onto the common control bus comprising:

determining at least one invalid command in used coding space of a coding standard;

determining unused coding space;

encoding the at least one invalid command in the used coding space and at least one command in the unused coding space.

8. A method as recited in claim 7 wherein the coding standard is an 8B10B (8 bit/10 bit) coding standard.

9. A method as recited in claim 8 wherein the invalid command is 111111.

10. A method as recited in claim 8 wherein the invalid command is 000000.

11. A method as recited in claims 9 or 10 wherein the invalid command occurs in a first six bits of the coding standard.

12. A method as recited in claims 9 or 10 wherein the invalid command occurs in a second bit through a seventh bit of the coding standard.

13. A method as recited in claims 9 or 10 wherein the invalid command occurs in a third bit through an eighth bit of the coding standard.

14. A method as recited in claims 9 or 10 wherein the invalid command occurs in a fourth bit through a ninth bit of the coding standard.

15. A method as recited in claims 9 or 10 wherein the invalid command occurs in a fifth bit through a tenth bit of the coding standard.

16. A method as recited in claim 6 for calibrating phases of the first data stream and the second data stream comprising:

a) choosing a phase;

b) testing to see if the phase is accurate;

c) receiving results of the testing;

d) logging the results of the testing;

e) repeating steps a) through d) for at least one more phase;

f) finding a threshold rate based on the results of the testing; and

g) dividing the threshold rate by two.

17. A disk drive controller comprising:

parallel logic developing parallel control signals;

serial logic developing serial control signals; and

a multiplexer coupling at least one of the parallel control signals and the serial control signals to a common bus.

18. A disk drive controller as recited in claim 17 further comprising:

one or more parallel hard disk drives coupled to the common bus and responsive to the parallel control signals; and

one or more serial hard disk drives coupled to the common bus and responsive to the serial control signals.

19. A hard disk drive controller as recited in claim 18 wherein the parallel hard disk drive is an ATA type.

20. A hard disk drive controller as recited in claim 18 wherein the serial hard disk drive is an SATA type.

21. A hard disk drive controller as recited in claim 18 wherein the serial control signals are sent to at least two of the one ore more serial hard disk drives at effectively double a base data rate.

22. A hard disk drive controller as recited in claim 21 wherein the doubling the base data rate comprises:

serial logic developing a sampling data clock;

serial logic developing a first data stream at the base data rate;

serial logic developing a second data stream at the base data rate; and

the multiplexer multiplexing the first data stream to the common control bus on a rising edge of the base data clock and the second data stream to the common control bus on a falling edge of the base data clock, whereby the common control

bus carries both the first data stream and the second data stream at effectively double the base data rate.

23. A hard disk drive controller as recited in claim 17 for encoding additional commands onto the common bus comprising:

determining at least one invalid command in used coding space of a coding standard;

determining unused coding space;

encoding the at least one invalid command in the used coding space and at least one command in the unused coding space.

24. A hard disk drive controller as recited in claim 23 wherein the coding standard is an 8B10B (8 bit/10 bit) coding standard.

25. A hard disk drive controller as recited in claim 24 wherein the invalid command is 111111.

26. A hard disk drive controller as recited in claim 24 wherein the invalid command is 000000.

27. A hard disk drive controller as recited in claims 25 or 26 wherein the invalid command occurs in a first six bits of the coding standard.

28. A hard disk drive controller as recited in claims 25 or 26 wherein the invalid command occurs in a second bit through a seventh bit of the coding standard.

29. A hard disk drive controller as recited in claims 25 or 26 wherein the invalid command occurs in a third bit through an eighth bit of the coding standard.

30. A hard disk drive controller as recited in claims 25 or 26 wherein the invalid command occurs in a fourth bit through a ninth bit of the coding standard.

31. A hard disk drive controller as recited in claims 25 or 26 wherein the invalid command occurs in a fifth bit through a tenth bit of the coding standard.

32. A hard disk drive controller as recited in claim 22 for calibrating phases of the first data stream and the second data stream comprising:

a) choosing a phase;

b) testing to see if the phase is accurate;

c) receiving results of the testing;

d) logging the results of the testing;

e) repeating steps a) through d) for at least one more phase; and

f) finding a best sampling pointer based on the results of the testing.

33. A method for doubling a data rate on a disk drive serial bus comprising:

developing a sampling data clock;

developing a first data stream at a base data rate;

developing a second data stream at the base data rate; and

multiplexing the first data stream to a disk drive serial bus on a rising edge of the base data clock and the second data stream to the disk drive serial bus on a falling edge of the base data clock, whereby the disk drive serial bus carries both the first data stream and the second data stream at effectively double the base data rate.

34. A method as recited in claim 33 for calibrating phases of the first data stream and the second data stream comprising:

a) choosing a phase;

b) testing to see if the phase is accurate;

c) receiving results of the testing;

d) logging the results of the testing;

e) repeating steps a) through d) for at least one more phase;

f) finding a threshold rate based on the results of the testing; and

g) dividing the threshold rate by two.

35. A method as recited in claim 33 for encoding additional commands onto the disk drive serial bus comprising:

determining at least one invalid command in used coding space of a coding standard;

determining unused coding space;

encoding the at least one invalid command in the used coding space and at least one command in the unused coding space.

36. A method as recited in claim 36 wherein the coding standard is an 8B10B (8 bit/10 bit) coding standard.

37. A method as recited in claim 36 wherein the invalid command is 111111.

38. A method as recited in claim 36 wherein the invalid command is 000000.

39. A method as recited in claims 37 or 38 wherein the invalid command occurs in a first six bits of the coding standard.

40. A method as recited in claims 37 or 38 wherein the invalid command occurs in a second bit through a seventh bit of the coding standard.

41. A method as recited in claims 37 or 38 wherein the invalid command occurs in a third bit through an eighth bit of the coding standard.

42. A method as recited in claims 37 or 38 wherein the invalid command occurs in a fourth bit through a ninth bit of the coding standard.

43. A method as recited in claims 37 or 38 wherein the invalid command occurs in a fifth bit through a tenth bit of the coding standard.

44. A method for encoding additional commands in a coding standard comprising:

determining at least one invalid command in used coding space of a coding standard;

determining unused coding space;

encoding the at least one invalid command in the used coding space and at least one command in the unused coding space.

45. A method as recited in claim 44 wherein the coding standard is an 8B10B (8 bit/10 bit) coding standard.

46. A method as recited in claim 45 wherein the invalid command is 111111.

47. A method as recited in claim 45 wherein the invalid command is 000000.

48. A method as recited in claims 46 or 47 wherein the invalid command occurs in a first six bits of the coding standard.

49. A method as recited in claims 46 or 47 wherein the invalid command occurs in a second bit through a seventh bit of the coding standard.

50. A method as recited in claims 46 or 47 wherein the invalid command occurs in a third bit through an eighth bit of the coding standard.

51. A method as recited in claims 46 or 47 wherein the invalid command occurs in a fourth bit through a ninth bit of the coding standard.

52. A method as recited in claims 46 or 47 wherein the invalid command occurs in a fifth bit through a tenth bit of the coding standard.